Crypto Bot: AI-Assisted Cryptography Scheme Selection

1.Introduction:

The field of cryptography plays a crucial role in ensuring the security and integrity of data and communication systems. Cryptographic primitives, such as block ciphers, hash functions, encryption schemes, and digital signatures, form the building blocks of modern cryptographic protocols and applications. However, selecting the appropriate cryptographic primitive for a specific use case can be a daunting task, especially for individuals with little or no background in cryptography.

Choosing the right cryptographic primitive requires a comprehensive understanding of various factors, including security requirements, performance considerations, compatibility with other cryptographic operations, and adherence to industry standards and best practices. Failure to select an appropriate primitive can lead to severe consequences, including data breaches, compromised communications, and potential legal and financial repercussions.

This project aims to address the challenge of selecting suitable cryptographic primitives by developing a software solution that leverages the capabilities of Artificial Intelligence (AI) systems. By combining the power of AI with a user-friendly interface, this project seeks to bridge the gap between cryptographic expertise and practical application, ensuring that even users with minimal cryptographic knowledge can make informed decisions regarding the selection of cryptographic primitives.

The software implementation incorporates two complementary AI systems: a decision tree-based system and a chatbot system. The decision tree-based system guides users through a structured series of prompts and multiple-choice or multiple-answer questions, providing recommendations based on their responses. On the other hand, the chatbot system leverages natural language processing capabilities to engage in open-ended conversations, catering to users who prefer a more conversational approach.

By integrating these AI systems with a comprehensive implementation usage table for various cryptographic primitives, the project aims to provide a seamless and intuitive experience for users, empowering them to make informed decisions and ensuring the secure and efficient implementation of cryptographic solutions across various applications and domains.

2. Cryptographic Primitive Selection

The selection of the appropriate cryptographic primitive is a crucial step in ensuring the security and effectiveness of any cryptographic system. For initial stage of this project, the chosen cryptographic primitive is block ciphers, which are symmetric-key encryption algorithms that operate on fixed-length blocks of plaintext data, transforming them into ciphertext blocks of the same length.

Block ciphers were selected as the focus of this project due to their widespread use and importance in various applications, such as file encryption, secure communication protocols, and data protection in databases and storage systems. They are essential components of many cryptographic protocols and are widely used in industries ranging from finance and healthcare to government and defense.

The decision to concentrate on block ciphers was also influenced by the complexity involved in selecting the appropriate scheme based on various factors. These factors include security requirements, performance considerations, compatibility with other cryptographic operations, and adherence to industry standards and best practices.

To address the challenges associated with block cipher selection, this project incorporates a comprehensive implementation usage table. This table outlines six widely used block cipher schemes: DES, Triple DES, Blowfish, AES, IDEA, and SIMON. The table captures essential features and properties of each scheme, such as:

- Standardization: Indicates the standardization status of the scheme, which is crucial for ensuring widespread adoption and compatibility.
- Runtime Efficiency: Provides a qualitative assessment of the scheme's runtime performance, enabling users to consider computational requirements and resource constraints.
- Security: Describes the known security vulnerabilities or strengths of the scheme, ensuring that users can make informed decisions based on their security requirements.
- Block Size: Specifies the fixed input and output block size of the scheme, which is essential for compatibility with other cryptographic operations and protocols.
- Key Size: Indicates the supported key sizes for the scheme, which directly impacts the overall security and resistance to cryptanalytic attacks.
- Type: Classifies the scheme based on its structure and design principles (e.g., Feistel, Substitution-Permutation Network), providing insights into the underlying cryptographic mechanisms.

By focusing on block ciphers and providing a comprehensive implementation usage table, this project aims to empower users with the necessary information and guidance to navigate the complex landscape of block cipher selection. The AI systems integrated into the software solution leverage this table to provide personalized recommendations based on users' specific requirements and preferences.

Furthermore, the inclusion of the SIMON scheme, a lightweight block cipher designed for resource-constrained devices, and the addition of a "Runtime Efficiency" column demonstrate the project's commitment to addressing a diverse range of use cases and staying up to date with the latest developments in the field of cryptography.

To cater to users who prefer a more open-ended and conversational approach, the Crypto Bot project incorporates a chatbot system that leverages the Google GEMINI AI system. By clicking the "CHAT" button on the main interface, users are directed to a chat interface where they can engage in natural language conversations with the GEMINI AI.

The GEMINI AI system has been fine-tuned using the implementation usage table, allowing it to provide more relevant and accurate responses related to block cipher schemes and their features. This integration enables users to ask open-ended questions, seek clarification, or explore alternative scenarios, facilitating a more flexible and personalized experience.

By combining the structured decision tree-based system and the conversational chatbot system, the Crypto Bot project caters to diverse user preferences and learning styles, ensuring that both guided and exploratory approaches are available for selecting the appropriate block cipher scheme.

3.Approach

The Crypto Bot project employs a multi-pronged approach to assist users in selecting appropriate block cipher schemes, even with minimal cryptographic expertise. This approach leverages the power of Artificial Intelligence (AI) systems, coupled with a comprehensive implementation usage table, to provide users with tailored recommendations and guidance.

Decision Tree-Based System:

Sample snippets of index.js, questions.js, picker.js

At the core of the project lies a decision tree-based system implemented using JavaScript code (picker.js and question.js). This system operates locally within the web browser, ensuring functionality even in scenarios where internet access is limited or unavailable. The decision tree structure is defined in the questionData object (question.js), where each node represents a question, answer choices, and a function to handle user responses.

The decision tree-based system operates by presenting users with a structured series of prompts and multiple-choice or multiple-answer questions rendered dynamically on the webpage. These questions are strategically designed to elicit information about the user's requirements, constraints, and preferences. Based on the user's responses, the corresponding function in questionData is executed, updating the progress bar, hiding, or displaying answer choices, and creating the next question or providing the final recommendation.

This systematic approach mimics the experience of consulting with a knowledgeable cryptography expert. Through a guided and interactive process, the decision tree-based system eliminates the need for users to have in-depth cryptographic expertise, empowering them to make informed decisions while ensuring the security and integrity of their data or communications.

Chatbot System:

Recognizing the diverse preferences and learning styles of users, the project incorporates a conversational chatbot system powered by Google's GEMINI AI. This system caters to users who prefer a more open-ended and exploratory approach to selecting block cipher schemes.

The GEMINI AI chatbot has been fine-tuned using the comprehensive implementation usage table, ensuring that its responses are tailored to the domain of block ciphers and their associated properties. Users can engage in natural language conversations with the chatbot, asking open-ended questions, seeking clarifications, or exploring alternative scenarios.

Screenshots of python code integrating Geminis api to our web application and training it with different prompts

By integrating the chatbot system into a local web application, users can seamlessly access the AI-powered assistant, facilitating a conversational and interactive experience. The chatbot's natural language processing capabilities allow it to understand and respond to user queries in a human-like manner, effectively bridging the gap between technical complexity and user-friendly guidance.

Implementation Usage Table:

At the heart of both the decision tree-based system and the chatbot system lies the meticulously crafted implementation usage table. This table serves as a comprehensive knowledge base, containing detailed information about various block cipher schemes and their properties.

The implementation usage table has been enhanced with the addition of a new column for "Runtime Efficiency" and a new row for the SIMON block cipher scheme. The "Runtime Efficiency" column provides a qualitative assessment of each scheme's runtime performance, enabling users to consider computational requirements and resource constraints. The inclusion of the **Blowfish** scheme, a lightweight block cipher designed for resource-constrained devices, demonstrates the project's commitment to addressing a diverse range of use cases and staying up to date with the latest developments in the field of cryptography.

Criteria / Scheme	Standardization	Runtime Efficiency	Security	Block Size	Key Size	Type
DES	Was a US standard	Was fast, now slower than AES	Breakable via efficient exhaustive key-search attack	64 bits, short for hash function uses	56 bits (+8 parity bits)	Feistel
Triple DES	Based on DES which was a standard	Slower than DES and AES	Not known to be efficiently breakable	64 bits, short for hash function uses	168 bits (three 56-bit DES keys)	Feistel
Blowfish	Intended as an alternative to the aging DES	A fast cipher that faster than AES	Vulnerable to birthday attack	64 bits, short for hash function uses	32 – 448 bits	Feistel
AES	Current US standard	Fast enough for almost all applications(except for resource-constrained devices)	Not known to be efficiently breakable	128 bits, large enough for hash function uses	128, 192, or 256 bits	SPN
IDEA	Proposed as a DES replacement. Patents were issued for it.	Fast enough for almost all applications(except for resource-constrained devices)	Not known to be efficiently breakable	64 bits, large enough for hash function uses	128 bits	Lai–Massey
SIMON	Wannabe standard	Faster than AES, targeting usage on resource-constrained devices	Breakable with large-computation attack	Short (32) to large (128)	64, 72, 96, 128, 144, 192 or 256 bits	Feistel

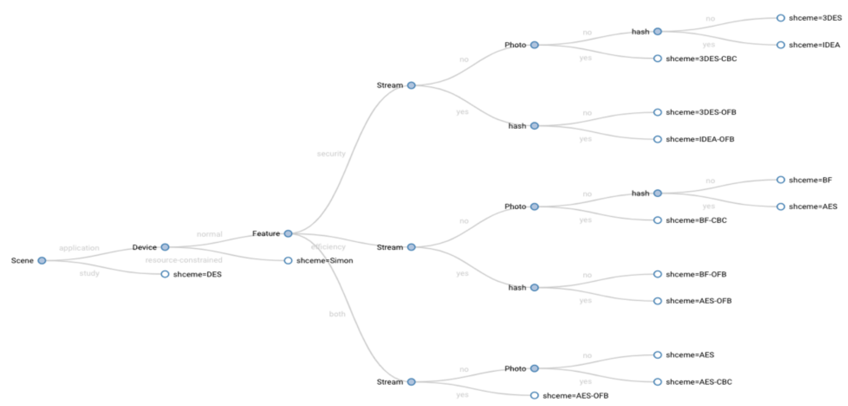
Update table.

Choice Criteria → es - Spring 2024 CS-GY 6903 I INET CS-UY 4783 A	andardization	Runtime Efficiency	Security	Block Length
DES	Was a US standard	Was fast, now slower than AES	Breakable via efficient exhaustive key-search attack	Short for hash function uses
Triple DES	Based on DES which was a standard	Slower than DES and AES	Not known to be efficiently breakable	Short for hash function uses
AES	Current US standard	Fast enough for almost all applications (except for resource-constrained devices)	Not known to be efficiently breakable	Large enough for hash function uses
Other ciphers: e.g., IDEA	Proposed as a DES replacement. Patents were issued for it.	Fast enough for almost all applications (except for resource-constrained devices)	Not known to be efficiently breakable	Large enough for hash function uses
Other ciphers: e.g., SIMON	Wannabe standard	Faster than AES, targeting usage on resource-constrained devices	Breakable with large-computation attack	Short (32) to large (128)

Old table

By leveraging this comprehensive implementation usage table, both the decision tree-based system and the chatbot system can provide users with informed recommendations based on various scheme properties, security considerations, and user requirements.

Through this multi-faceted approach, the Crypto Bot project empowers users of all expertise levels to navigate the complex landscape of block cipher selection. By offering both structured guidance and open-ended exploration, the project caters to diverse user preferences, ensuring a tailored and effective experience in selecting the most appropriate block cipher scheme for their specific needs.



Decision tree

4. AI System Implementation

The Crypto Bot project leverages the power of AI systems to facilitate the process of selecting appropriate block cipher schemes. To achieve this, a Python codebase was developed to integrate with the AI system's APIs and perform two crucial tasks:

a. Training the AI System

The first step in the implementation process involved training the AI system with a labeled dataset derived from the comprehensive implementation usage table. This table, meticulously crafted and enhanced with additional columns and rows, served as the foundation for the AI system's knowledge base.

The Python code read the table data and preprocessed it into a format suitable for training the AI system. This involved converting the table entries into labeled examples, where each row represented a specific block cipher scheme, and the columns represented the corresponding features and properties.

By ingesting this labeled dataset, the AI system could learn to associate the various block cipher schemes with their respective characteristics, such as standardization status, runtime efficiency, security considerations, block size, key size, and cipher type.

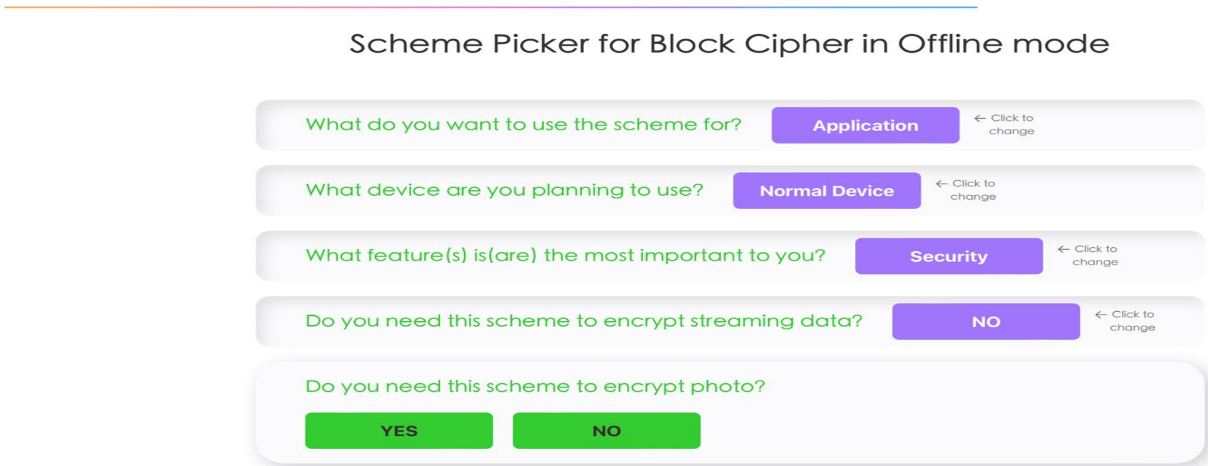
b. Generating Adaptive Prompts and Recommending Schemes

The second crucial task accomplished by the Python codebase was to calculate a sequence of prompts for the AI system to interact with the user. These prompts were designed to be adaptive, meaning that the subsequent prompts were generated based on the user's previous responses and the AI system's answers.

The code implemented a decision tree-like structure, where each prompt corresponded to a node in the tree. Depending on the user's response to a prompt, the code would traverse the tree and generate the next appropriate prompt. This adaptive approach ensured that the prompts were tailored to the user's specific requirements and preferences, leading to a more personalized and efficient experience.

The prompts were presented to the user in the form of multiple-choice or multiple-answer questions, allowing for a user-friendly and intuitive interaction. To cater to non-expert users, the code also considered scenarios where the user might not know the answer to one or more prompts. In such cases, the AI system could provide guidance or suggest alternative paths to continue the selection process.

After the user had answered all the prompts, the Python code instructed the AI system to analyze the collected responses and recommend a valid block cipher scheme from the learned implementation usage table. Additionally, the system could provide a ranked list of schemes, allowing the user to explore alternative options if desired.



5. Evaluation and Success Metrics

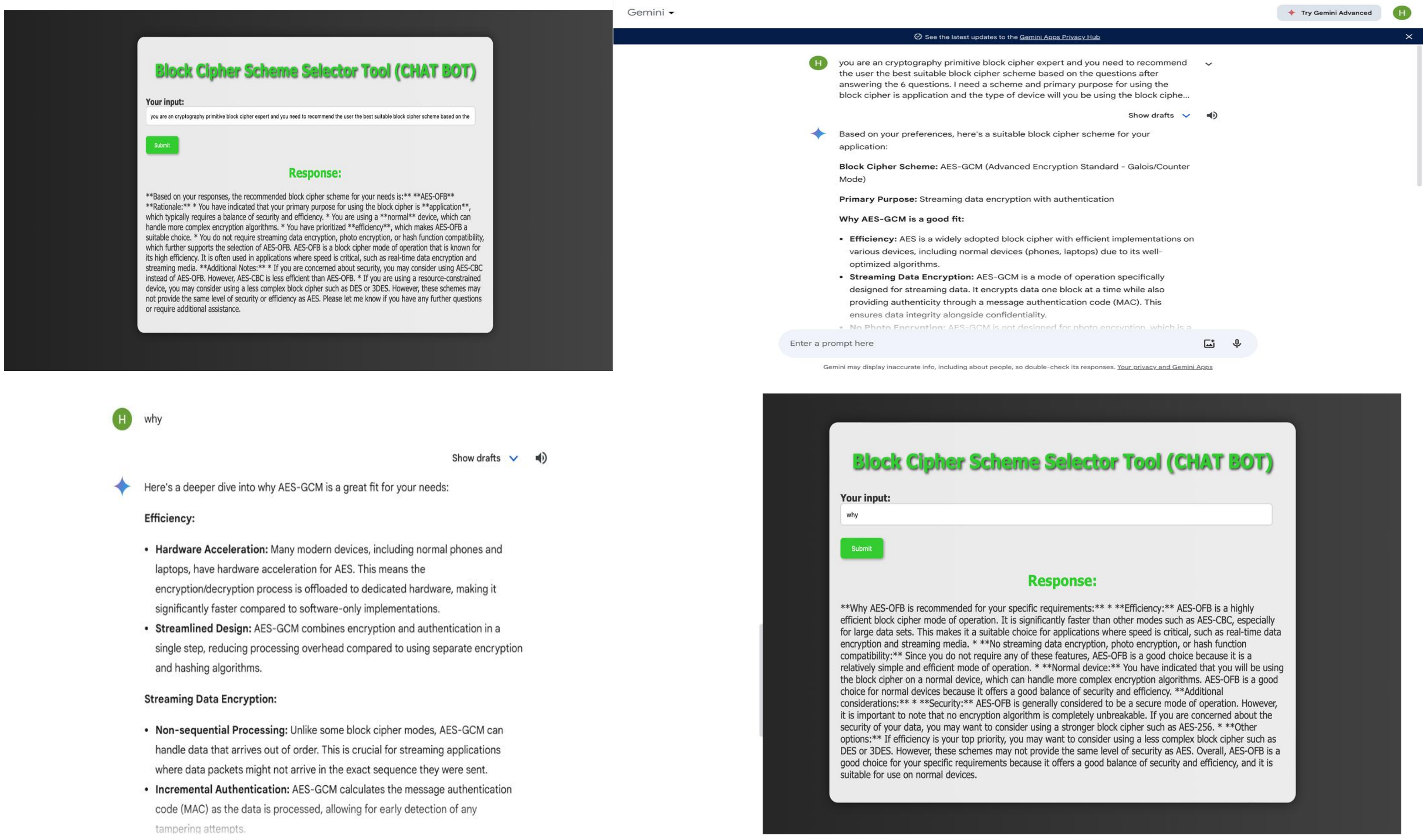
To evaluate the effectiveness of the Crypto Bot's approach, a meaningful success metric was employed: the choice correctness success probability. This metric was calculated through a series of experiments, each following a specific procedure:

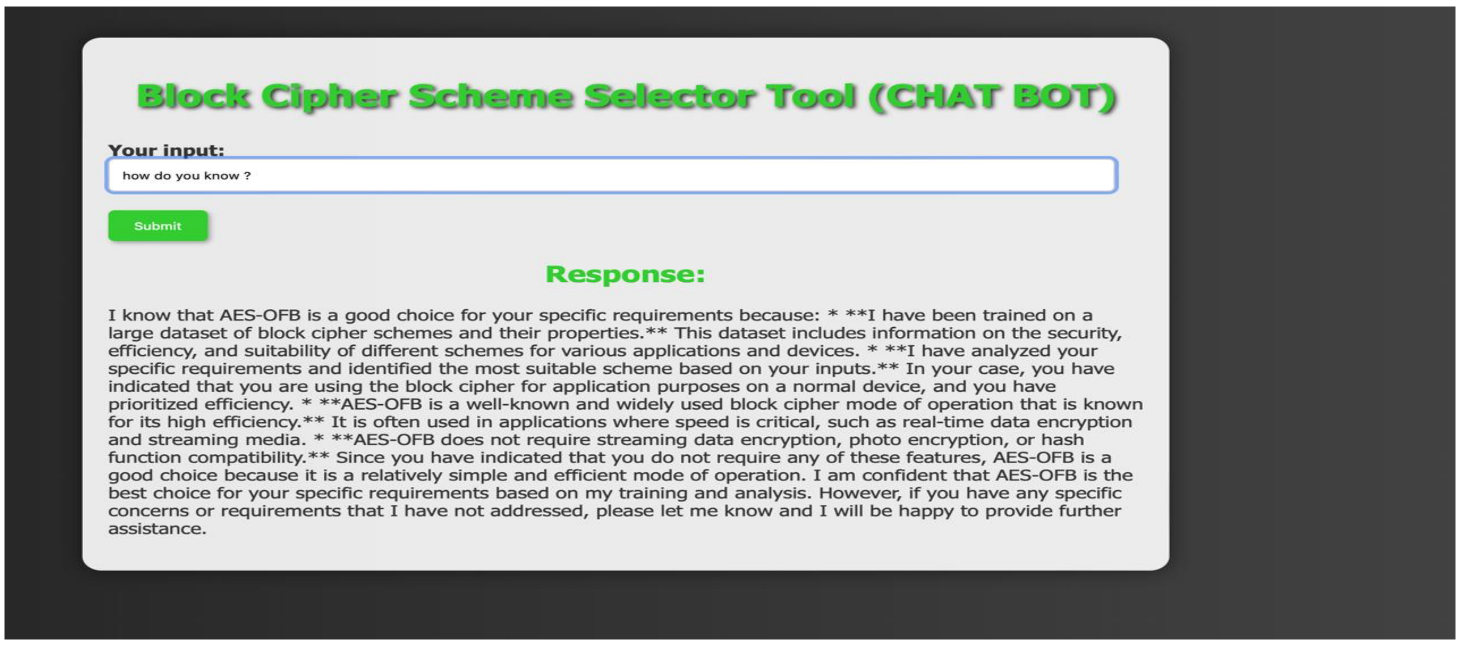
- Randomly choose a block cipher scheme from the implementation usage table.
- Interact with the AI system by providing responses consistent with the chosen scheme's features and properties.
- Record whether the AI system correctly recommended the chosen scheme (correct choice) or provided an incorrect recommendation (incorrect choice).

By repeating this experiment multiple times and aggregating the results, the choice correctness success probability could be calculated as the ratio of correct choices to the total number of experiments.

The evaluation process aimed to assess the AI system's ability to accurately recommend the appropriate block cipher scheme based on the user's responses. A high choice correctness success probability would indicate that the Crypto Bot's approach, leveraging the implementation usage table and the adaptive prompting system, was effective in guiding users to the correct block cipher scheme selection.

Furthermore, the evaluation process also provided insights into potential areas for improvement, such as refining the implementation usage table, optimizing the decision tree structure, or enhancing the AI system's training process to better capture the nuances of block cipher scheme selection.





6.Conclusion:

The Crypto Bot project successfully implemented a Python codebase to integrate with AI system APIs, enabling the training of the AI system using a labeled dataset derived from the comprehensive implementation usage table. This training phase equipped the AI system with the necessary knowledge to understand the relationships between block cipher schemes and their associated features and properties.

Furthermore, the Python code facilitated the generation of adaptive prompts, presented as multiple-choice or multiple-answer questions to the user. The prompts were tailored based on the user's previous responses, creating a personalized and efficient experience. The AI system's ability to recommend valid block cipher schemes, including ranked lists of options, was a key outcome of this implementation.

To evaluate the effectiveness of the Crypto Bot's approach, a rigorous evaluation process was conducted, calculating the choice correctness success probability through a series of experiments. This metric provided valuable insights into the system's performance and highlighted areas for potential improvement, further enhancing the project's ability to assist non-expert users in selecting appropriate block cipher schemes.

By combining a comprehensive implementation usage table, AI system integration, adaptive prompting, and robust evaluation methods, the Crypto Bot project demonstrated its potential to bridge the gap between cryptographic expertise and practical application, empowering users to make informed decisions in the realm of secure data transmission and storage.

References:

https://csrc.nist.gov/csrc/media/events/lightweight-cryptography-workshop-2015/documents/papers/session1-shors-paper.pdf

https://eprint.iacr.org/2016/625.pdf

https://ctf101.org/cryptography/what-are-block-ciphers/

https://www.freecodecamp.org/news/what-is-a-block-cipher/

https://www.splashdata.com/splashid/blowfish.html